UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/903,019	07/10/2001	Murari Kumar	42390P11769	9066

8791 7590 03/31/2005

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
CHOW, CHIH CHING	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 03/31/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/903,019	KUMAR, MURARI	
	Examiner	Art Unit	
	Chih-Ching Chow	2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 November 2004*.

2a)☒ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-28* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-28* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 November 2004* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to amendment dated November 19, 2004.

Response to Amendments

2. The objection to drawings for FIG. 2 is withdrawn in view of Applicants' amendment to specification dated 11/19/2004.

3. The objections to drawings for FIG. 11 is maintained since 'device executable' is still very similar to 'device' even the device executables in 512 and 514 are formed in accordance with the embodiments described in Applicant's specification. The diagram should be marked 'Prior Art' since the 'device executable' is still very similar to Fout's page 6, Figure 1.

4. The objection to claim 2 is maintained since only one 'service-control' is changed to 'service control' in view of Applicants' amendment to specification dated 11/19/2004; there are two more in the following paragraphs.

5. The objection to claim 19 is maintained since 'compiling the service-control class files' (in 5th paragraph) is not changed to service control in view of Applicants' amendment to specification dated 11/19/2004.

6. Applicant's arguments for Claims 1-28 have been fully considered respectfully by the examiner but they are not persuasive. See MPEP 7.38 Arguments are Moot because of new ground(s) of rejection Applicant's arguments with respect to claims 1, 2, 10, 19, 23-28 (new) have been considered but are moot in view of the new ground(s) of rejection. The 'generating class files according to a UPnP device description document' and 'to provide an implementation of a UPnP

network protocol' were never mentioned in the original claims, therefore a new prior art is referenced in this office action.

Claim Objections

7. Claims 23-28 are objected to because of the following informalities: 'service control stub method' is referenced in claims 23-28, however 'service-control stub-method' is used though out the entire document, claims 23-28 should follow the previous references. Appropriate correction is required.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

9. Claims 1-6, 8-15, 17-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 2002/0035621, by Zintel et al. (hereinafter "Zintel"), in view of US Patent No. 6,789,077, by Gregory L. Slaughter et al. (hereinafter "Slaughter").

Claims	**Zintel / Slaughter**
1. (Currently Amended) A method comprising: ~~receiving a UPnP device desciption document form a device developer;~~	Zintel teaches using desription document, see Zintel paragraph 10, "The description is expressed in XML and includes **vendor-specific**

generating one or more service control class files including one or more service-control stub-methods according to a UPnP device description document received from a device developer;

receiving the service control class files including ~~updated~~ the service-control stub-methods ~~modified~~ updated by the device developer for responding to actions and events received by a UPnP device described by the UPnP device description document; and

compiling the service-control class files and the updated service-control stub-methods along with a device class library and a UPnP software development kit to generate a UPnP device executable to provide an implementation of a UPnP network protocol for the UPnP device described by the UPnP device description document.

manufacturer information like the model name and number, serial number, manufacturer name, URLs to vendor-specific Web sites, etc. The description also includes a list of any embedded devices or services, as well as URLs for control, eventing, and presentation." Zintel teaches using UPnP device description document expressed in XML, but he doesn't mention 'generating new code (*executable*) base on description' specifically, however, Slaughter teaches it in an analogous prior art, see Slaughter column 18, lines 62-67, "the distributed computing environment differs from other distributed computing environments in that instead of passing the necessary code between objects necessary to access the other object, the environment provides access to XML descriptions of an object or target so that **code may be generated based on the XML description** to access the target." Also see Slaughter column 24, lines 51-56, "A generation tool may be provided for the pre-construction of gates. The generation tool may include an XML **parser**, a code generator and a code compiler. In one embodiment, the code generator may be a Java source code generator and the code compiler may be a Java code compiler." When generating code it may very likely generating new control class files, and

more service-control stub-methods. -- It all depends on what language it is implemented with.
It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement of Zintel's disclosure of using UPnP device description document with generating new device executable base on an XML description taught by Slaughter for the purpose of ensuring that the generated gate code is bug-free (see Slaughter column 22, lines 57-64).

2. (Currently Amended) The method of claim 1, wherein generating the ~~service-control~~ service control class files further comprises:

parsing the UPnP device description document to determine a root device including one or more services and one or more embedded device each including one or more services, each service defined by a service control protocol description (SCPD) file;

generating a service-control class file for each of the one or more services of the root device;

selecting an embedded device from the one or more embedded devices of the root device;

generating a service-control class file for each of the one or more services of the selected embedded device;

repeating the generating and selecting for each of the one or more embedded

For the features of claim 1 see claim 1 rejection. For the rest of claim 2 feature, see Zintel paragraph 89, "There is **exactly one SCPD per Service Definition**. SCPDs adhere to XML grammar. **SCPDs can be generated by an automated tool** that accepts a SST Definition and a Command Set Definition as input." Zintel also teaches 'root device' in paragraph 71, "**Devices can contain other Devices**. UPnP enables the association of user interface (display icon and root Web page) with every Device, including Root Device." and all the service control class file should be disseminate to the sub-devices, see paragraph 701, "**repeat once for each embedded device** defined by a UPnP Forum working committee. If UPnP vendor differentiates device by embedding additional UPnP devices,

for each of the one or more embedded devices of the root device; and

once each of the service control class files are generated, generating a class service linker file based on the generated service-control classes, the class service file linker enabling linking the service-control classes during compiling to generate the device executable.

repeat once for each embedded device." The service class files are able to be linked, see Zintel paragraph 547, "Both the Announce and Discovery packets also **contain a link** or a URL to an XML file that is used by the embedded computing device at the device description phase to describe itself".

3. (Original) The method of claim 2, further comprising:

generating a root directory for the root device;

storing each of the one or more services of the root device within the root directory;

selecting an embedded device from the one or more embedded devices of the root device;

generating a sub-root directory for the selected embedded device;

storing each of the one or more services of the selected embedded device within the sub-root directory; and

repeating generating and storing for each of the one or more embedded devices of the root device

For the features of claim 2 see claim 2 rejection. Zintel has disclosed the rest of claim 3 feature in paragraph 532, "UPnP also provides a Directories mechanism to allow discovery to scale...When present, a **directory** will read all incoming **service** requests and respond to them itself. This requires that all **services** (e.g., the **embedded computing device** 900) register with the **directory** so that the **directory** is able to properly answer on their behalf. The **directory** is also responsible for communicating with other directories in order to determine whether the service is available within the local network, the WAN and potentially the Internet." Items b-f apply to claim2 rejection (***for all nested devices***).
It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement of the Weisman, Spring, and UPnP's disclosure with the directory mechanism taught by Zintel

for the purpose of when introducing a new device into a network to automatically configure so as to connect and interact with other computing devices available on the network. (see Zintel Abstract, lines 5-7).

4. (Original) The method of claim 2, wherein generating service control class files for the root device further comprises:

selecting a service from the one or more services of the root device;

generating a class generator object for the selected service of the root device;

passing an SCPD file describing the selected service, including a device identification code of the root device and a service identification of the selected service to the class generator object;

generating, by the class generator object, a service control class file based on the received SCPD file, the service identification code and the device identification code;

generating a header file corresponding to the generated service control class file and the SCPD file;

once generated, destroying the class generator object; and

repeating the selecting, generating, passing, generating and destroying for each of the one or more services of the

For the feature of claim 2 see claim 2 rejection. For the device identifier feature see Zintel paragraph 69, "The formal definition of a Device Type. A Device Definition includes a **Device Type Identifier**, the fixed elements in the Description Document, the required set of Service Definitions in the Root Device, and the hierarchy of required Devices and Service Definitions." For the 'header file' see Zintel paragraph 201, "SSDP specifies Service Type (ST), Notification type (NT), and Unique Service Name (USN) header fields for queries and for announcements. **UPnP uses the ST or NT header to carry one of the UPnP defined identifiers.** A unique USN is required for each unique SSDP announcement."

root device.

5. (Original) The method of claim 2, wherein generating service control class files for the selected embedded device further comprises: selecting a service from the one or more services of the selected embedded device; generating a class generator object for the selected service of the selected embedded device; passing an SCPD file describing the selected service, including a device identification code of the selected embedded device and a service identification code of the selected service to the class generator object; generating, by the class generator object, a service control class file based on the received SCPD file, the service identification code and the device identification code; generating a header file corresponding to the generated service control class file and the SCPD file; once generated, destroying the class generator object; and repeating the selecting, generating, passing, generating and destroying for each of the one or more services of the selected embedded device.	Same as claim 2 rejection.
6. (Original) The method of claim 2, wherein generating the service linker class file further comprises:	Same as claim 2 and claim 3 rejections.

generating a class service linker object;

storing class information regarding each generated service control class file within a service table of the class service linker object;

generating, by the class service linker object, the class service linker file based on the service table; and

destroying the class service linker object.

8. (Original) The method of claim 1, further comprising:

executing the device executable;

creating an instance of the root device and each of the one or more services of the root device;

creating an instance of each embedded device and each of the one or more services of the respective embedded device;

organizing the root device and embedded devices, as well as the services of the root device and the embedded devices within a tree hierarchy based on the device description document to form a device object tree; and

registering the root device and the one or more embedded device within the device object tree with the device class library to enable receipt of events for the services of the root device and the services of the one or more registered embedding devices.

Same as claim 1 rejection.

9. (Original) The method of claim 8, wherein registering the root device and one or more embedded devices further comprises:

registering an event listener object of the device class library with the UPnP software development kit to enable receipt of action and event requests received/generated by one or more control points of the UPnP device;

registering the root device and one or more embedded devices with the event listener object;

receiving, by the event listener object, a respective action/event request from a control point;

finding a service object for response to the respective action request using a received device identification code and service identification code;

once the service object is found, invoking a callback function of the service object to determine an appropriate action method to execute in response to the respective action request;

executing the appropriate action method; and

once the action method is processed, setting an event object with a response string that is received by the control point.

Same as claim 8 rejection.

10. (Currently Amended) A computer readable storage medium including

Same as claim 1 rejection.

program instructions that direct a computer to function in a specified manner when executed by a processor, the program instructions comprising:

~~receiving a UPnP device description document form a device developer;~~

generating one or more service control class files including one or more service-control stub-method according to a UPnP device description document received from a device developer;

receiving the service control class files including ~~updated~~ the service-control stub-methods ~~modified~~ updated by the device developer for responding to actions and events received by a UPnP device described by the UPnP device description document; and

compiling the service-control class files and the updated service-control stub-methods along with a device class library and a UPnP software development kit to generate a UPnP device executable to provide an implementation of a UPnP network protocol for the UPnP device.

11. (Original) The computer readable storage medium of claim 10, wherein generating the service-control class files further comprises:

parsing the UPnP device description document to determine a root device including one or more services and one or more embedded device each including one or more

See claim 10 and claim 2 rejections.

services, each service defined by a service control protocol description (SCPD) file;

generating a service-control class file for each of the one or more services of the root device;

selecting an embedded device from the one or more embedded devices of the root device;

generating a service-control class file for each of the one or more services of the selected embedded device;

repeating the generating and selecting for each of the one or more embedded devices of the root device; and

once each of the service control class files are generated, generating a class service linker class file based on the generated service-control classes, the class service linker file enabling linking the service-control classes during compiling to generate the device executable.

12. (Original) The computer readable storage medium of claim 11, further comprising:

generating a root directory for the root device;

storing each of the one or more services of the root device within the root directory;

selecting an embedded device from the one or more embedded devices of the root device;

See claim 11 and claim 3 rejections.

generating a sub-root directory for the selected embedded device;

storing each of the one or more services of the selected embedded device within the sub-root directory; and

repeating generating and storing for each of the one or more embedded devices of the root device.

13. (Original) The computer readable storage medium of claim 11, wherein generating service control class files for the root device further comprises:

selecting a service from the one or more services of the root device;

generating a class generator object for the selected service of the root device;

passing an SCPD file describing the selected service, including a device identification code of the root device and a service identification of the selected service to the class generator object;

generating, by the class generator object, a service control class file based on the received SCPD file, the service identification code and the device identification code;

generating a header tile corresponding to the generated service control class file and the SCPD file;

once generated, destroying the class generator object; and

repeating the selecting, generating,

See claim 11 and claim 4 rejections.

passing, generating and destroying for each of the one or more services of the root device.

14. (Original) The computer readable storage medium of claim 11, wherein generating service control class files for the selected embedded device further comprises:

selecting a service from the one or more services of the selected embedded device;

generating a class generator object for the selected service of the selected embedded device;

passing an SCPD file describing the selected service, including a device identification code of the selected embedded device and a service identification code of the selected service to the class generator object;

generating, by the class generator object, a service control class file based on the received SCPD file, the service identification and the device identification;

generating a header file corresponding to the generated service control class file and the SCPD file;

once generated, destroying the class generator object; and

repeating the selecting, generating, passing, generating and destroying for each of the one or more services of the selected embedded device.

See claim 11 and claim 5 rejections.

15. (Original) The computer readable storage medium of claim 11, wherein generating the service linker class file further comprises:

generating a class service linker object;

storing class information regarding each generated service control class file within a service table of the class service linker object;

generating, by the class service linker object, the service class linker file based on the service table; and

destroying the class service linker object.

See claim 11 and claim 6 rejections.

17. (Original) The computer readable storage medium of claim 10, further comprises:

executing the device executable;

creating an instance of the root device and each of the one or more services of the root device;

creating an instance of each embedded device and each of the one or more services of the respective embedded device;

organizing the root device and embedded devices, as well as the services of the root device and the embedded devices within a tree hierarchy based on the device description document to form a device object tree; and

registering the root device and one or more embedded devices of the

See claim 10 and claim 8 rejections.

device object tree with the device class library to enable receipt of actions/events for the one or more services of the root device and the one or more services of the one or more registered embedded devices.

18. (Original) The computer readable storage medium of claim 17, wherein registering the root device and one or more embedded devices further comprises:

registering an event listener object of the device class library with the UPnP
software development kit to enable receipt of action and event requests received/generated by one or more control points of the UPnP device;

registering the root device and one or more embedded devices with the event listener object;

receiving, by the event listener object, a respective action/event request from a control point;

finding a service object for response to the respective action request using a received device identification code and service identification code;

once the service object is found, invoking a callback function of the service object to determine an appropriate action method to execute in response to the respective action request;

executing the appropriate action

See claim 17 and claim 9 rejections.

method; and

once the action method is processed, setting an event object with a response string that is received by the control point.

19. (Currently Amended) ~~An apparatus~~ system, comprising:

a processor having circuitry to execute instructions;

a communications interface coupled to the processor, the communications interface to advertise services to a control point, provide device description to the control point, provide service description for each service to the control point, to receive action/event requests from the control point and to publish updates during state changes in response to received action/event requests; and

a storage device coupled to the processor, having sequences of instructions stored therein, which when executed by the processor cause the processor to:

receive the service control class files including ~~updated~~ service-control stub-methods ~~modified~~ updated by the device developer for responding to actions and events received by a UPnP device described by the UPnP device description document,

compile the service-control class files and the updated service-control stub-methods along with a device class

See claim 1 and claim 9 rejections.

library and a UPnP software development kit to generate a UPnP device executable to provide an implementation of a UPnP network protocol for the UPnP device described by the UPnP device description document, and

execute the UPnP device executable to enable response to actions/events received by the UPnP device.

20. (Original) The apparatus of claim 19, wherein the instruction to execute the device executable further causes the processor to:

create an instance of a root device and each of one or more services of the root device;

create an instance of each embedded device of the root device and each of one or more services of a respective embedded device;

organize the root device and embedded devices, as well as the services of the root device and the embedded devices within a tree hierarchy based on the device description document to form a device object tree;

register the device object tree with the device class library to enable receipt of actions/events for the services of the root device and the services of the embedded devices; and

register an event listener class of the device class library with the UPnP

See claim 19 and claim 8 rejections.

software developing kit to receive action/event requests from one or more control points of the UPnP device that are forwarded to service objects within the device object tree.

21. (Original) The apparatus of claim 19, comprising:

one or more root devices, each including one or more services for responding to actions and events received by the respective root device, and one or more embedded devices including one or more services to respond to actions and events received by a respective embedded device.

See claim 19 and claim 8 rejections.

22. (Original) The apparatus of claim 20, wherein the processor is further caused to:

receive a UPnP device description document from a device developer,

generate one or more service control class files including one or more service-control stub-method based on the UPnP device description document and one or more service control protocol description files listed within the UPnP device description document, and

provide the service control class files including service-control stub-methods to the device developer in order to receive updated service control stub-methods including code for responding to actions and events received by a

See claim 20 and claim 8 rejections.

UPnP device described by the UPnP device description document.	
23. (New) A method comprising: generating at least one service control class file including at least one service control stub method according to a UPnP description document received from a device developer; and generating a UPnP device executable for a UPnP device described by the UPnP device description document wherein the device executable to provide an implementation of a UPnP network protocol for the UPnP device.	See claim 1 rejection.
24. (New) The method of claim 23 wherein generating the device executable further comprises: receiving the service control class files including the service control stub methods updated by the device developer for responding to actions and events received by the UPnP device described by the UPnP device description document; and compiling the service control class files and the updated service control stub methods along with a device class library and a UPnP software development kit to generate the UPnP device executable.	For the feature of claim 23 see claim 23 rejection, for the rest of claim 24 feature see claim 1 rejection.

10. Claims 7, 16, and 25-28 are rejected under 35 U.S.C. 103(a) as being unpatentable over US 2002/0035621, by Zintel et al. (hereinafter "Zintel"), in view

of US Patent No. 6,789,077, by Gregory L. Slaughter et al. (hereinafter "Slaughter"), further in view of 'Universal Plug and Play in Windows XP' by Tom Fout (hereinafter "Fout").

CLAIM	Zintel / Slaughter / Fout
7. (Original) The method of claim 1, wherein receiving the service control class files further comprises: displaying the one or more service control stub-methods to the device developer; receiving code from the device developer for implementing the one or more service control stub-methods for responding to actions and events received by the UPnP device; and once the code is received, storing the received code within the one or more corresponding service control stub-methods.	For the features of claim 1 see claim 1 rejection. Zintel and Slaughter have taught all the aspects but they don't mention the 'displaying the service to the device developer'. However, Fout has taught these features in an analogous art, in Fout, page 19, under 'Presentation', last paragraph "The capabilities of the presentation page (***presentation means 'present' to the user, it implies to be 'displayed' to the user***) are completely specified by the UPnP vendor. To implement a presentation page, a UPnP vendor may wish to use UPnP mechanisms for control and/or events, leveraging the device's existing capabilities. Notice that there is no UPnP Forum element defined in presentation, it is completely up to the vendor!" In UPnP, a 'representation' template is shown on page 15-16, and on page 22, under 2.4 'Description: UPnP Service Template', "By appropriate specification of placeholders, the listing above can be either a UPnP Device Template or a UPnP device description. Recall that some placeholders would be defined by a UPnP Forum working committee (colored *red*), i.e., the UPnP device type identifier, required UPnP services, and required UPnP embedded devices (if any). If these were defined,

the listing would be a UPnP Device Template, **codifying** the standard for this type of device. UPnP Device Templates are one of the key deliverables from UPnP Forum working committees.· Here we've learned that a template can be displayed for the code developer to define the UPnP device. The entered code is received and stored for later use. It would have been obvious to a person of ordinary skill in the art at the time of the invention was made to supplement Weisman, Spring and UPnP's teaching by displaying, receiving and storing the device taught by Fout for the purpose of standardize device (see Fout, page 22, 8th paragraph).

16. (Original) The computer readable storage medium of claim 10, wherein receiving the service control class files further comprises:

displaying the one or more service control stub-methods to the device developer;

receiving code from the device developer for implementing the one or more service control stub-methods for responding to actions and events received by the UPnP device; and

once the code is received, storing the received code within the one or more corresponding service control stub-methods.

For the features of claim 10 see claim 10 rejection. For the rest of the features in claim 16, see claim 7 rejection.

25. (New) The method of claim 23, wherein generating the service control class file further comprises:

displaying the service control stub method to the device developer;

receiving code from the device developer for implementing the service control stub method for responding to actions and events received by the UPnP device; and

storing the code received from the device developer within the service control stub method.

For the features of claim 23 see claim 23 rejection. For the rest of the features in claim 25, see claim 7 rejection.

26. (New) A method comprising:

displaying, to a device developer, at least one service control stub method of at least one service control file generated according to a UPnP device description document received from the device developer; and

compiling the service control class files and the service control stub methods updated by the device developer along with a device class library and a UPnP software development kit to generate a UPnP device executable to provide an implementation of a UPnP network protocol for the UPnP device.

Same as claim 7 rejection.

27. (New) The method of claim 26, wherein displaying the at least one service control stub method further comprises:

receiving code from the device

For the feature of claim 26 see claim 26 rejection, for the rest of claim 27 feature see claim 3 rejection.

developer for implementing the service control stub method for responding to actions and events received by the UPnP device; and
storing the updated code received from the device developer within the service control stub method.

28. (New) The method of claim 26, wherein display further comprises:
receiving the UPnP device description document from the device developer; and
generating one or more service control class files including one or more service-control stub-methods according to the UPnP device description document received from the device developer.

For the feature of claim 26 see claim 26 rejection, for the rest of claim 28 feature see claim 1 and claim 2 rejections.

Allowable Subject Matter

11. The following is a statement of reasons for the indication of allowable subject matter:

Zintel et al. teaches all the features claimed in claims 1-28. However Zintel et al. when taken individually or in combination fails teach or suggest a method for generating executable based on description document as recited in claims 1, 10, 19, 23 and 26 is designed for a vendor rather than for a customer.

12. Claims 1, 10, 19, 23, and 26 would be allowable if rewritten to specify that the claimed invention is designed for the vendors when developing new devices for UPnP environment.

Conclusion

13. The following summarizes the status of the claims:

35 USC § 103 rejection: Claims 1-28

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

UPnP : 'UPnP Device Architecture', June 2000, a detailed introduction for UPnP architecture, which includes the Addressing, Discovery, Description, Control, Eventing and Presentation phases.

15. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will

the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chih-Ching Chow whose telephone number is 571-272-3693. The examiner can normally be reached on 7:00am - 3:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Dam can be reached on 571-272-3695. Any inquiry of a general nature or relating to the status of this applications should be directed to the **TC2100 Group receptionist**: 571-272-2100. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Chih-Ching Chow

Examiner

Art Unit 2122

CC

ANTONY NGUYEN-BA
PRIMARY EXAMINER